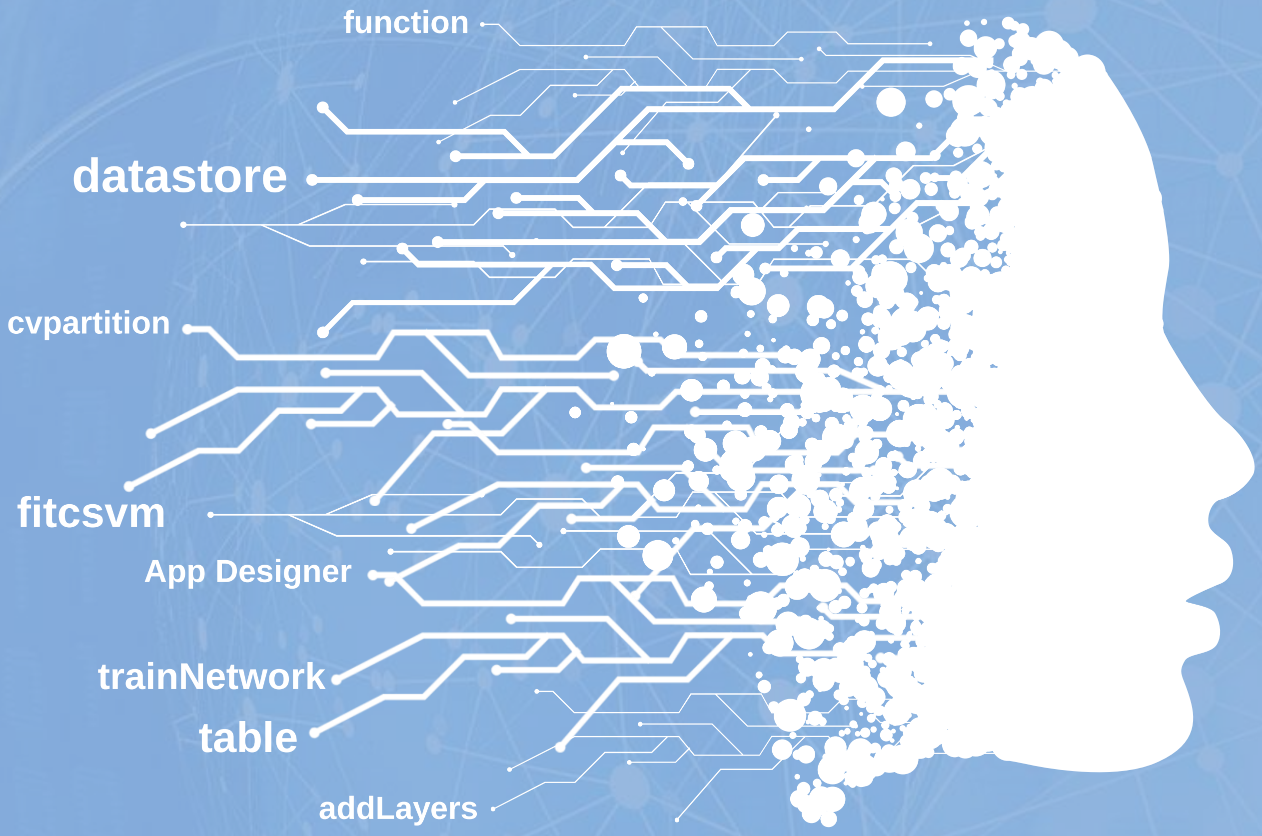


Classification: Support Vector Machine

MATLAB進階程式語言與實作

盧家鋒 Chia-Feng Lu, Ph.D.
Department of Biomedical Imaging and Radiological Sciences, NYCU
alvin4016@nycu.edu.tw

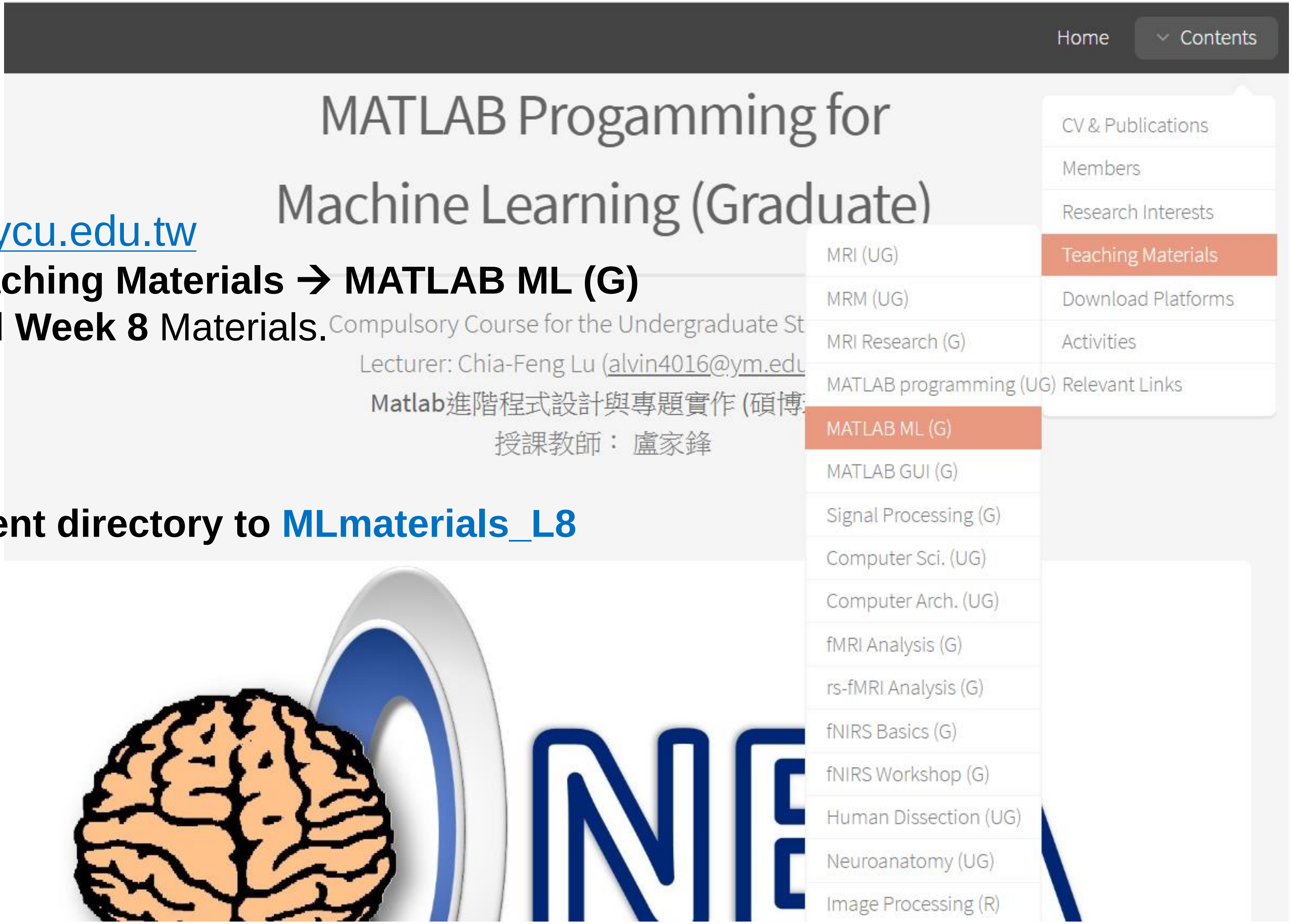
Teaching Materials

http://cflu.lab.nycu.edu.tw

Contents → Teaching Materials → MATLAB ML (G)

Please download **Week 8** Materials.

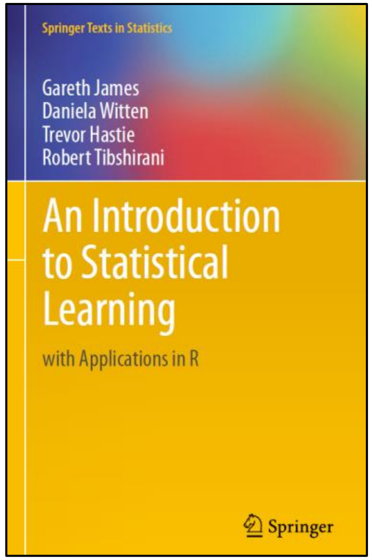
Please set current directory to **MLmaterials_L8**



References

[Textbook 3]

- **An Introduction to Statistical Learning, 2nd edition, 2013**
  Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani
- **Online resources:** https://github.com/rghan/ISLR
- **Online resources:** https://github.com/JWarmenhoven/ISLR-python
- **Support Vector Machines (Ch.9)**



Support Vector Machines

Basic concepts

Background

- Support vector machines (SVM) have been shown to perform well in a variety of settings, and are often considered one of the best classifiers.
- The SVM is a generalization of a simple and intuitive classifier called the maximal margin classifier.

Maximal margin classifier	⇨	Support vector classifier	⇨	Support vector machine
Linearly separable classes		An extension version for soft margin		Non-linear class boundaries
Hard margin		**Soft margin**		**Transform function & Kernel trick**

Hyperplane

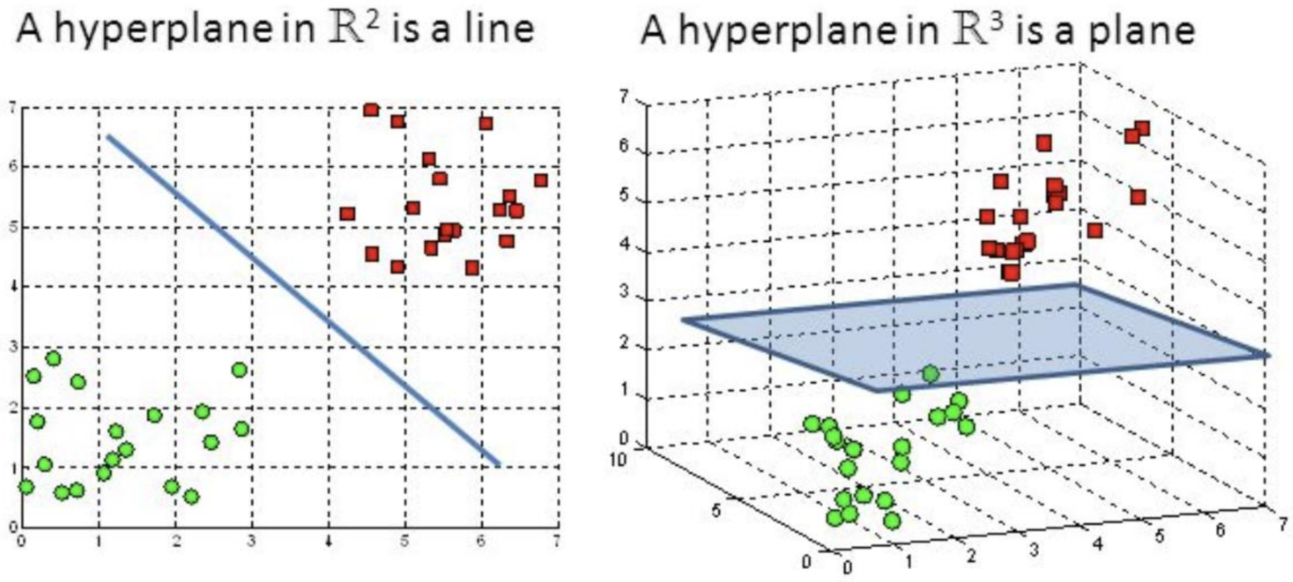
- **Hyperplane**: (p-1)-dimensional flat subspace

$$w^T x + b = 0$$

or

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0$$

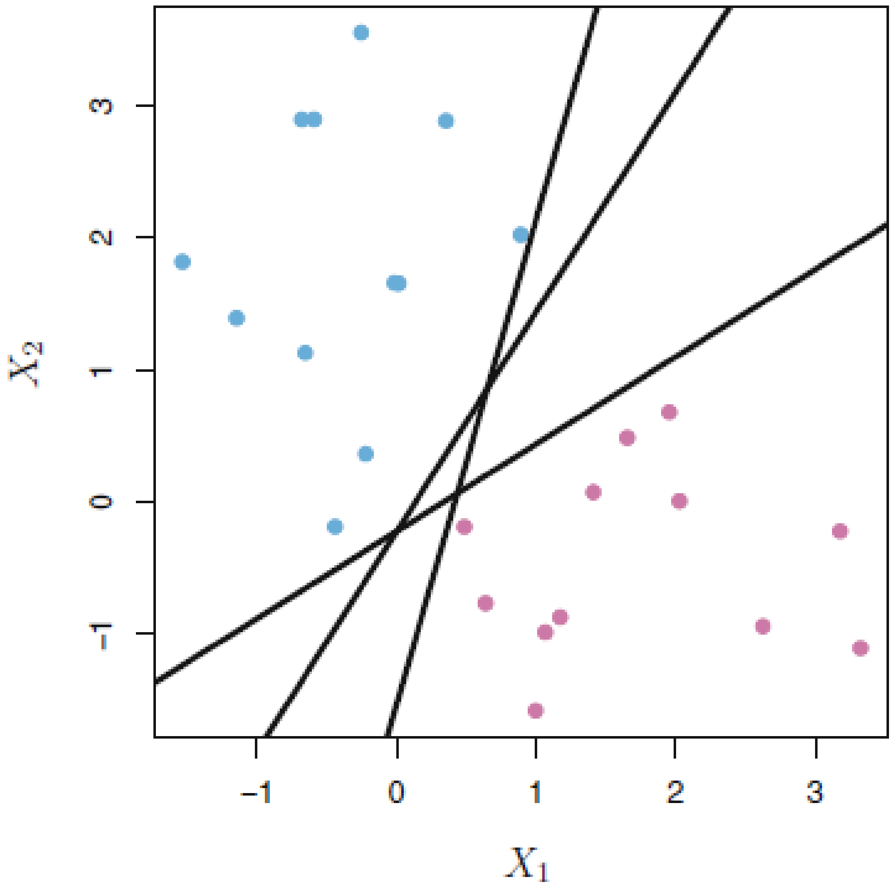
- **Separating hyperplanes**
  - Separates the training observations according to their class labels.



https://hackmd.io/@fairytien/support-vector-machine

Separating hyperplanes

- If our data can be perfectly separated using a hyperplane, then there will in fact exist an infinite number of such hyperplanes.
- This is because a given separating hyperplane can usually be shifted a tiny bit up or down, or rotated, without coming into contact with any of the observations.



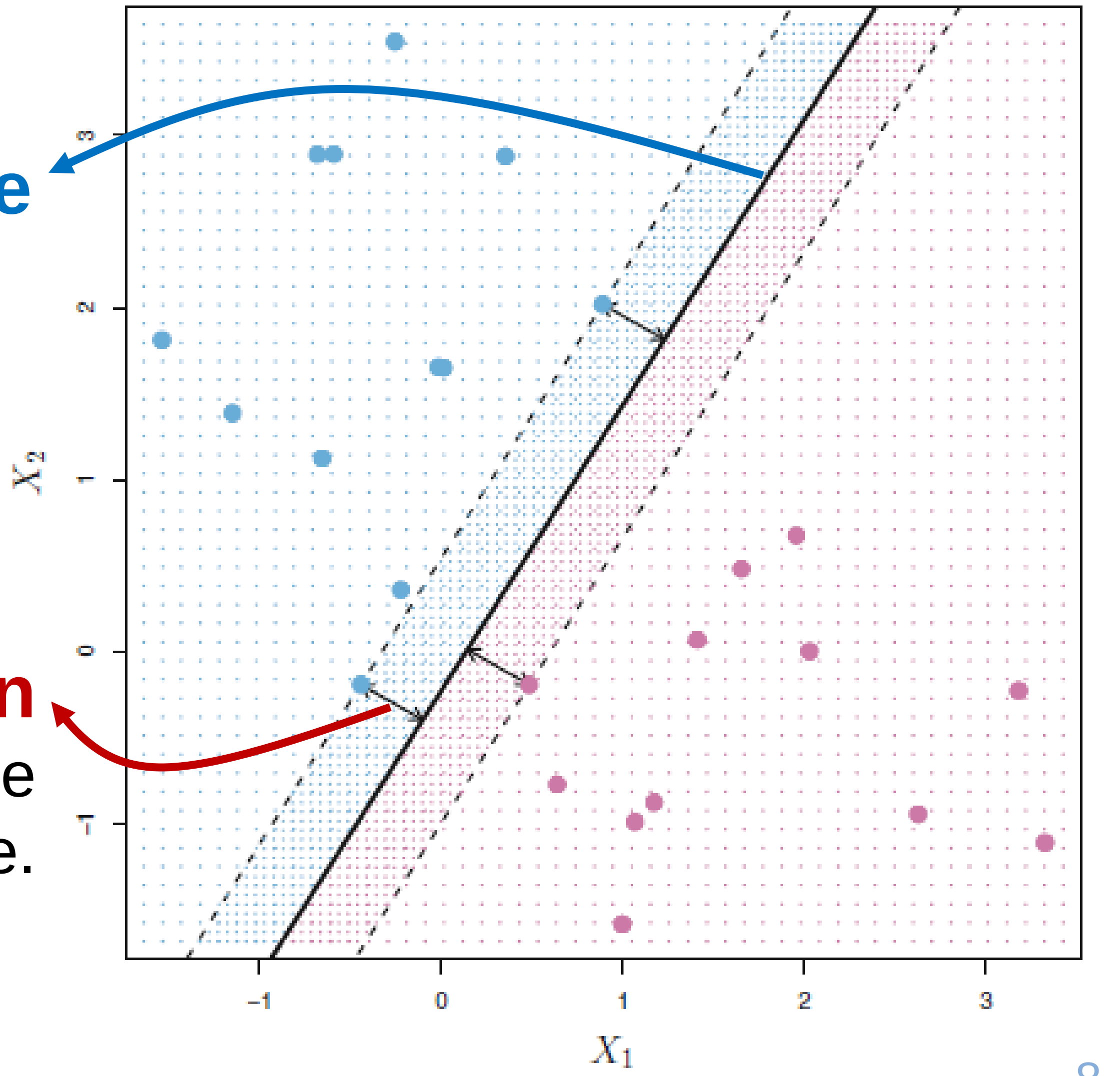
Maximal Margin Hyperplane

- **Optimal separating hyperplane**

  The one that is farthest from the closest training observations.

- **Margin**

  The perpendicular distance from the closest observations to the hyperplane.

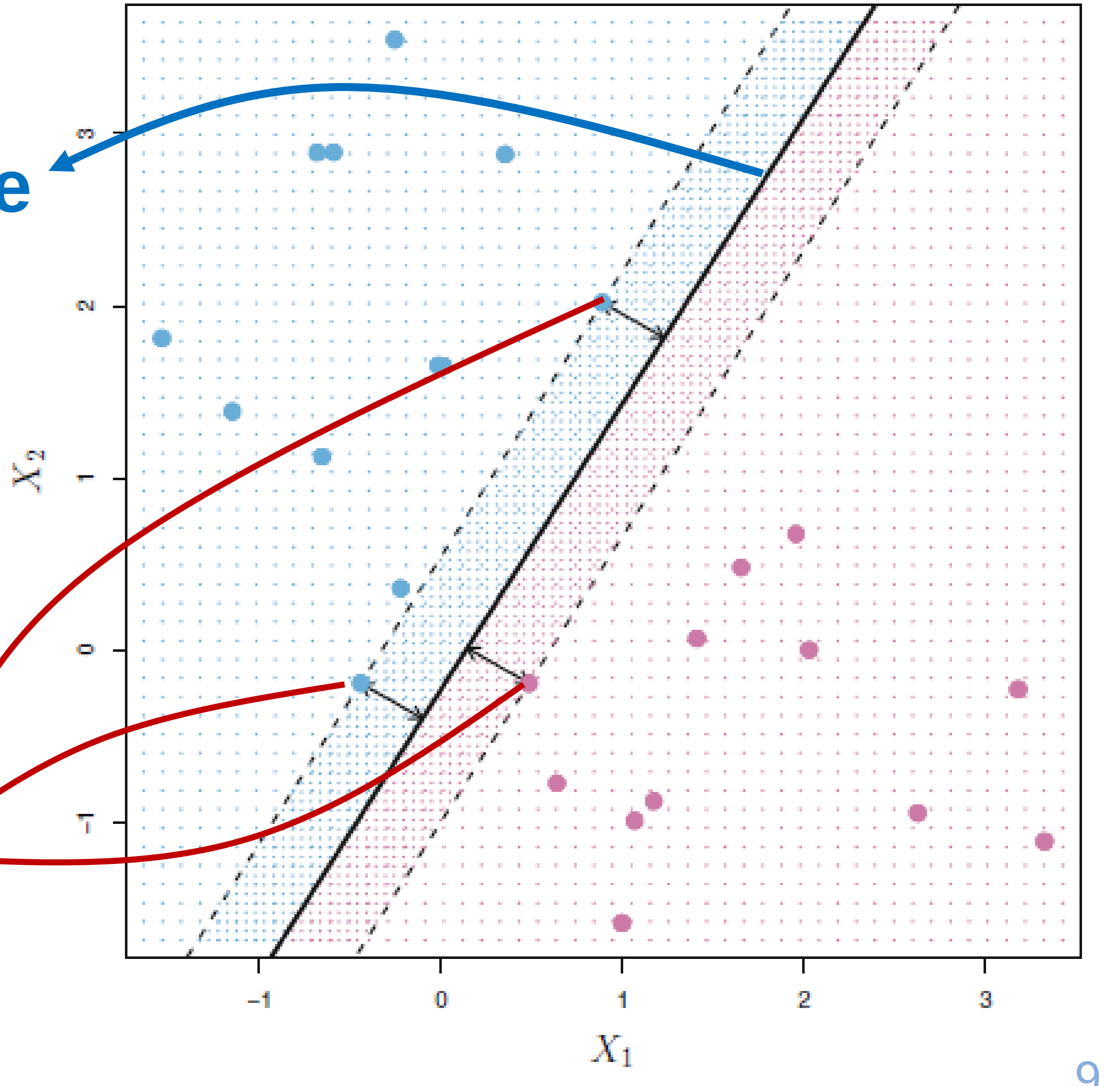


Maximal Margin Hyperplane

- **Optimal separating hyperplane**

  The one that is farthest from the closest training observations.

- In a sense, the maximal margin hyperplane represents the mid-line of the widest “slab” that we can insert between the two classes.



Support vectors

Maximal Margin Classifier

- Constructing the maximal margin hyperplane using **n** training observations:

$$\boldsymbol{x_1, \dots, x_n} \in \mathbb{R}^{\boldsymbol{p}}$$ p dimensional data sets

with associated class labels

$$\boldsymbol{y_1, \dots, y_n} \in \{-\mathbf{1}, \mathbf{1}\}$$ binary classification

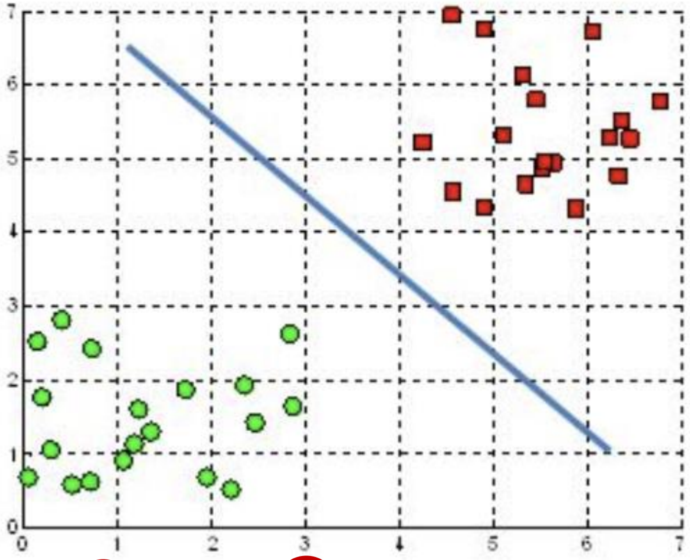
Maximal Margin Classifier

- The maximal margin hyperplane is the solution to the optimization problem.

Each observation is on the correct side of the hyperplane and at least a distance *M* from the hyperplane.

$$\underset{\beta_0,\beta_1,\ldots,\beta_p}{\text{maximize}}\; M$$

subject to $\displaystyle\sum_{j=1}^{p} \beta_j^2 = 1$

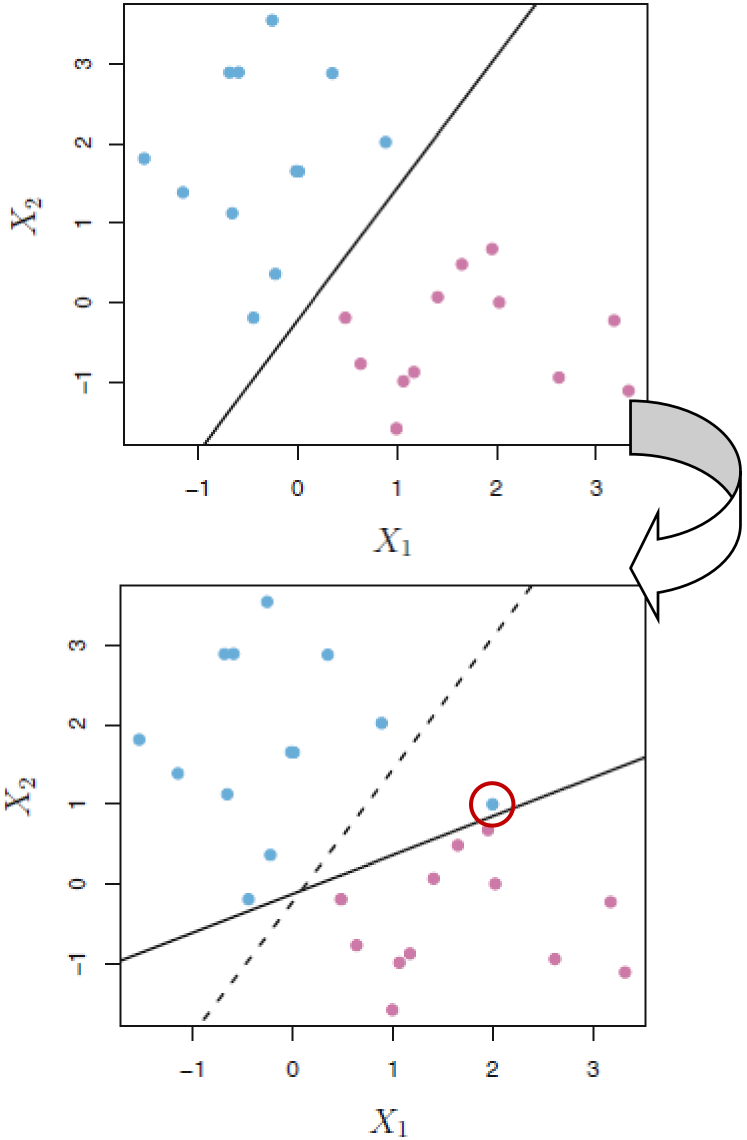


$$\underline{y_i}\big(\underline{\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}}\big) \geq M,\ \forall\, i = 1, \ldots, n$$

Maximal Margin Classifier

- A classifier based on a separating hyperplane will necessarily perfectly classify all of the training observations; this can lead to sensitivity to individual observations.

- They "support" the maximal margin hyperplane in the sense vector that if these points were moved slightly then the maximal margin hyperplane would move as well.

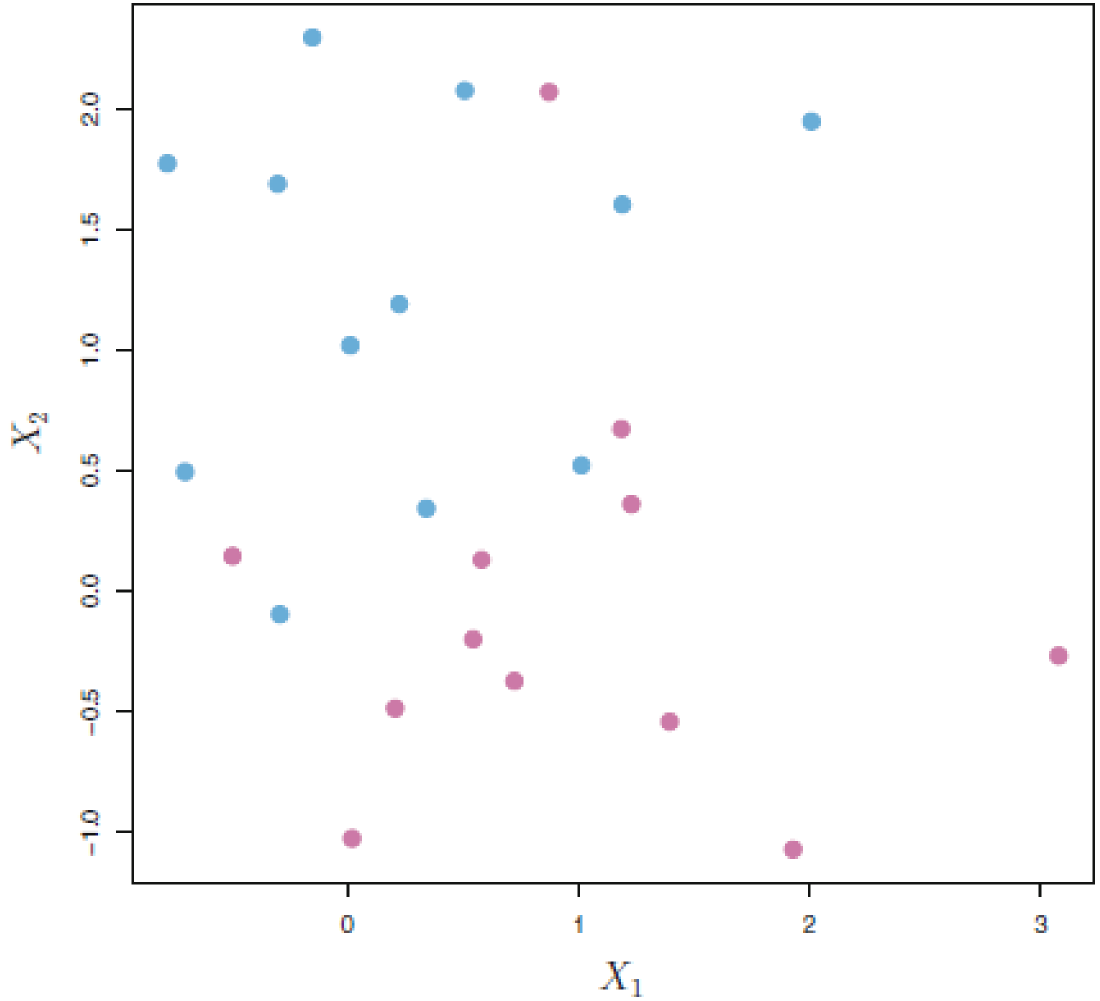


Non-separable Case

- In many cases no separating hyperplane exists, and so there is no maximal margin classifier (no solution with M>0).

- we can extend the concept of a separating hyperplane in order to develop a hyperplane that almost separates the classes, using a so-called soft margin.

Support vector classifier



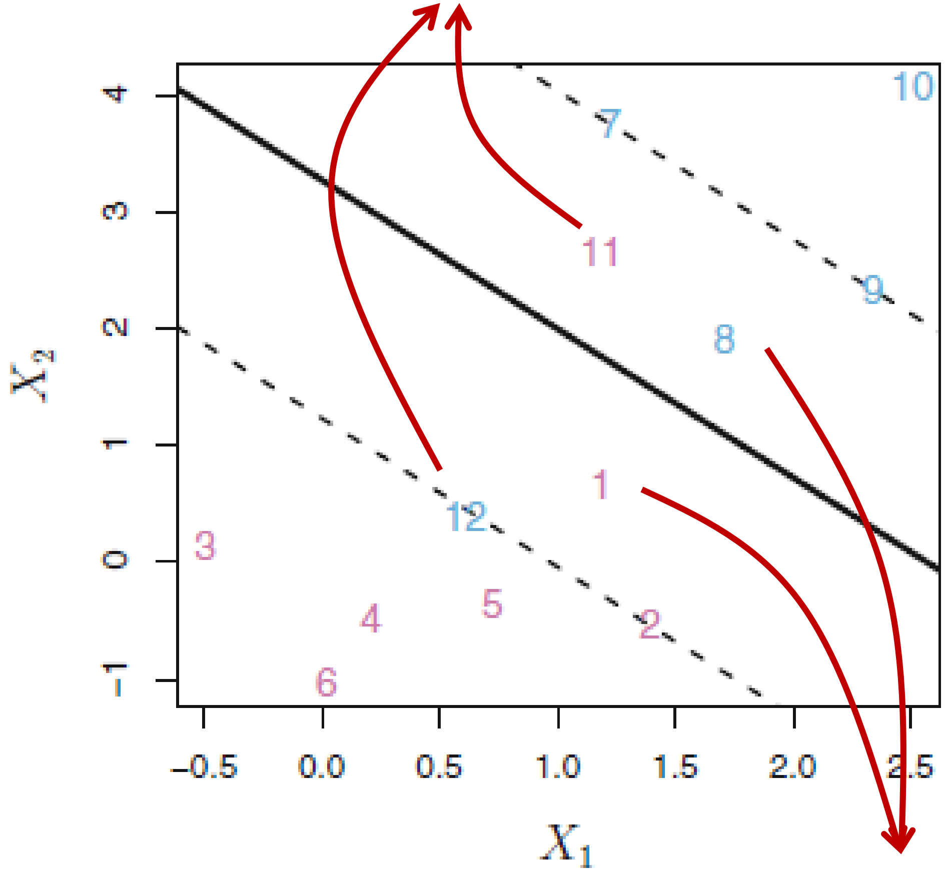
(Soft margin) Support Vector Classifier

Support vector classifier or **soft margin classifier**

- Greater robustness to individual observations
- Better classification of *most* of the training observations.

- We allow some observations to be on the incorrect side of the margin, or even the incorrect side of the hyperplane.

On the wrong side of the hyperplane and the margin



On the wrong side of the margin

(Soft margin) Support Vector Classifier

- The hyperplane is chosen to correctly separate most of the training observations into the two classes, but may misclassify a few observations.

$$\underset{\beta_0,\beta_1,\ldots,\beta_p,\epsilon_1,\ldots,\epsilon_n}{\text{maximize}} \quad M$$

p features and n observations

subject to $\sum_{j=1}^{p} \beta_j^2 = 1$

Slack variable

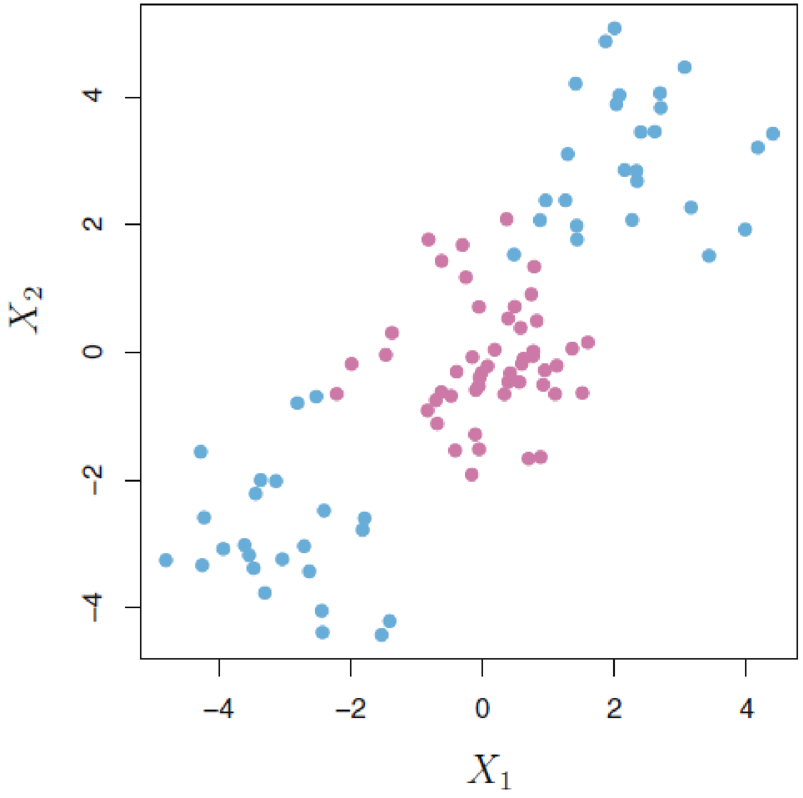
$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) \geq M(1 - \underline{\epsilon_i})$$

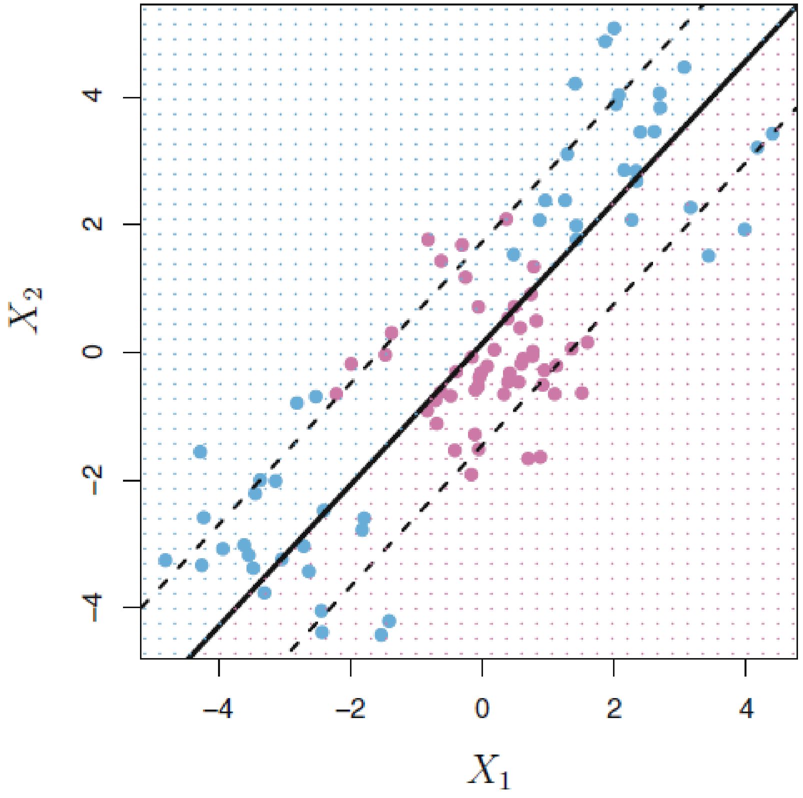
$$\underline{\epsilon_i \geq 0,} \sum_{i=1}^{n} \epsilon_i \leq C$$

C is a nonnegative tuning parameter (soft margin factor)

Non-linear Decision Boundaries

- It is clear that a support vector classifier or any linear classifier will perform poorly here (right-hand panel).





Non-linear Decision Boundaries

- In the case of non-linear regression, we consider enlarging the feature space using functions of the predictors, such as quadratic and cubic terms.

- we could fit a support vector classifier using **2*p*** features

$$\underset{\beta_0,\beta_{11},\beta_{12},\ldots,\beta_{p1},\beta_{p2},\epsilon_1,\ldots,\epsilon_n}{\text{maximize}} \quad M$$

p features and n (i=1,…,n) observations

Quadratic terms

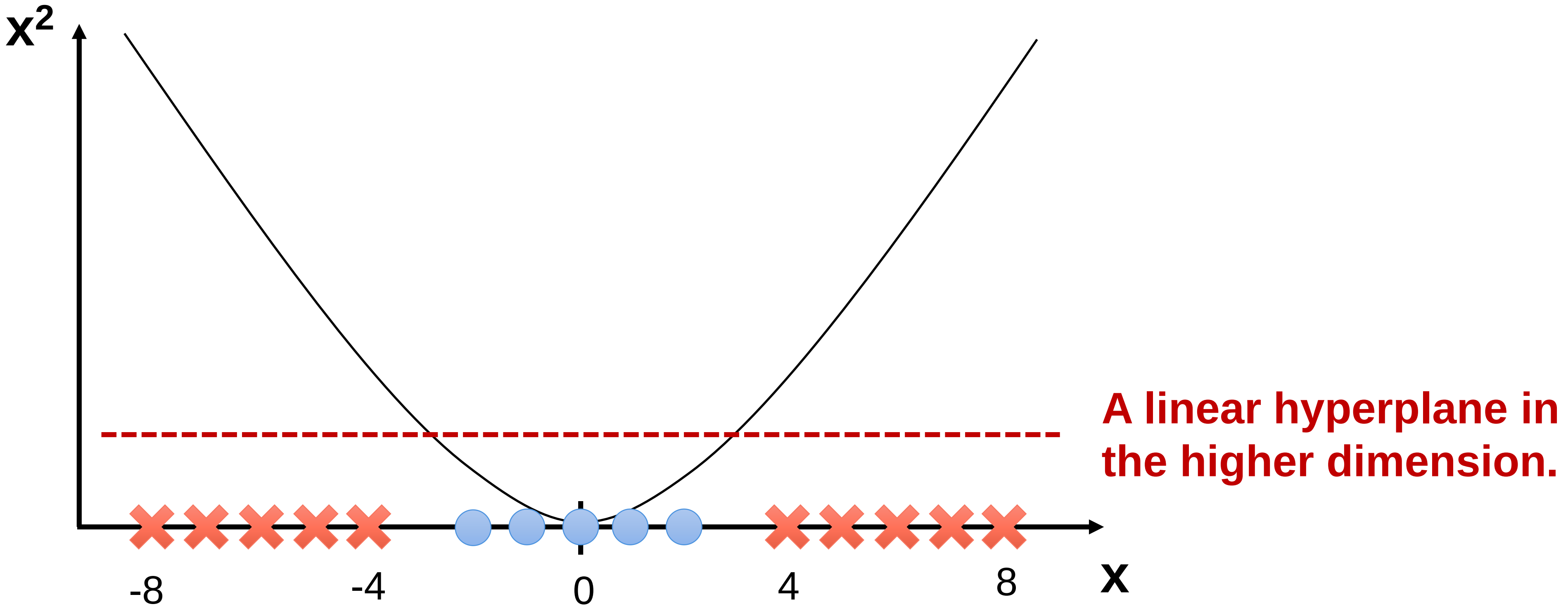
$$y_i\left(\beta_0+\sum_{j=1}^{p}\beta_{j1}x_{ij}+\sum_{j=1}^{p}\beta_{j2}x_{ij}^2\right)\geq M(1-\epsilon_i)$$

$$\epsilon_i\geq 0,\sum_{i=1}^{n}\epsilon_i\leq C,\quad \sum_{j=1}^{p}\sum_{k=1}^{2}\beta_{jk}^2=1$$

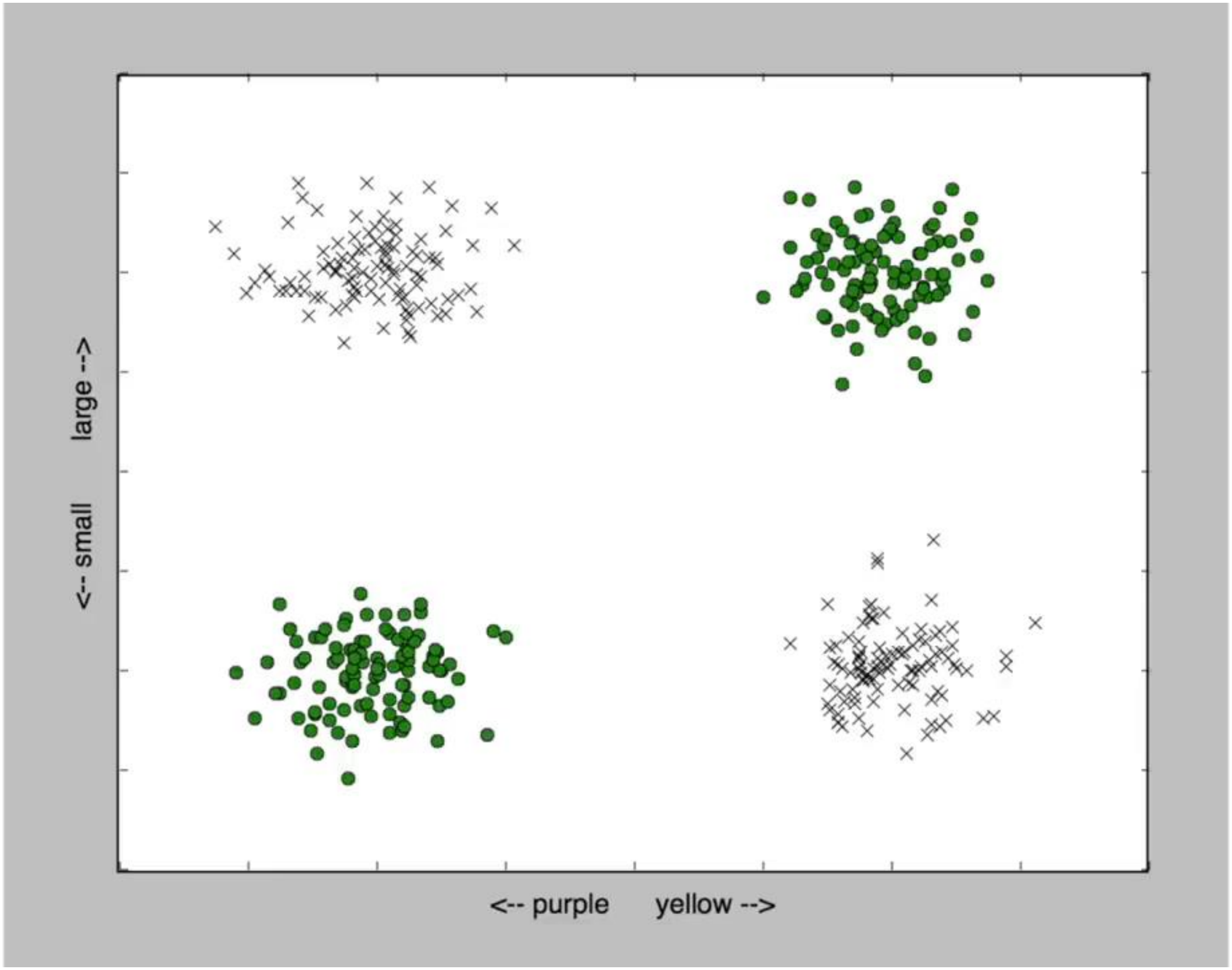
Support Vector Machine (SVM)

- A non-linear **transform/projection function**

$$\boldsymbol{\phi(x) = (x, x^2)}$$



Support Vector Machine (SVM)



https://youtu.be/-Z4aojJ-pdg

Solution to the linear SVM

- The solution function is

$$f(x) = \beta_0 + \sum_{i \in S} \alpha_i \langle x, x_i \rangle$$

$$\underset{\beta_0, \beta_1, \ldots, \beta_p, \epsilon_1, \ldots, \epsilon_n}{\text{maximize}} \quad M$$

subject to $\sum_{j=1}^{p} \beta_j^2 = 1$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) \geq M(1 - \epsilon_i)$$

$$\epsilon_i \geq 0, \sum_{i=1}^{n} \epsilon_i \leq C$$

Collection of support vectors

Lagrangian multiplier, the weights of the support vectors.
(0 for all non-support vectors)

Inner product

The inner product after transform

Ex:

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \ \phi(\boldsymbol{x}) = \begin{bmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{bmatrix}$$

Project to the higher dimension
($\mathbb{R}^2 \rightarrow \mathbb{R}^3$)

$$\langle x_m, x_n \rangle = x_{m1}x_{n1} + x_{m2}x_{n2}$$

2 multiplies + 1 additions

$$\Downarrow$$

Increasing computational costs
of inner product in the higher dimension!

$$\langle \phi(x_m), \phi(x_n) \rangle = x_{m1}^2 x_{n1}^2 + 2x_{m1}x_{m2}x_{n1}x_{n2} + x_{m2}^2 x_{n2}^2$$

10 multiplies + 2 additions

Kernel Trick

$$\langle \phi(x_m), \phi(x_n) \rangle = x_{m1}^2 x_{n1}^2 + 2x_{m1}x_{m2}x_{n1}x_{n2} + x_{m2}^2 x_{n2}^2$$

$$= (x_{m1}x_{n1} + x_{m2}x_{n2})^2 = \langle x_m, x_n \rangle^2$$

We only need to calculate the inner product in the original feature dimension!

- Useful kernel functions:

Linear: $\langle x_m, x_n \rangle$

Polynomial: $(\langle x_m, x_n \rangle + 1)^d$

Gaussian radial basis function: $e^{-\|x_m - x_n\|^2}$

Heart Dataset Including data from 303 patients.

- **Age**
- **Sex**
- **Chest Pain:** typical angina, atypical angina, non-anginal pain, asymptomatic
- **Rest BP:** resting blood pressure
- **Chol:** serum cholestoral in mg/dl
- **FBS:** fasting blood sugar > 120 mg/dl (1: true; 0: false)
- **Rest ECG:** 0: normal;1: having ST-T wave abnormality; 2: showing probable or definite left ventricular hypertrophy by Estes' criteria
- **Max HR:** maximum heart rate achieved
- **ExAng:** exercise induced angina (1: yes; 0: no)
- **Oldpeak:** ST depression induced by exercise relative to rest
- **Slope:** the slope of the peak exercise ST segment
  1: upsloping; 2: flat; 3: downsloping
- **Ca:** number of major vessels (0-3) colored by flourosopy
- **Thal:** Thalium stress test **(**normal, fixed defect, reversable defect)

AHD (the predicted attribute, diagnosis based on angiography)
(Yes: ≥ 50% diameter narrowing; No: < 50% diameter narrowing;)

MLmaterials_L8\Heart.csv

Exercise – Classification Tree

- Should remove patients with missing data.
- Perform cvpartition to hold out 30% data.

- Predict “AHD" (Yes or No)

```
predictors={'Age', 'Sex', 'ChestPain', 'RestBP', 'Chol', 'Fbs', 'RestECG',
'MaxHR', 'ExAng', 'Oldpeak', 'Slope', 'Ca', 'Thal'};
svm_model = fitcsvm(dataTrain,'AHD','PredictorNames',predictors,...
    'OptimizeHyperparameters',‘all',...
    'HyperparameterOptimizationOptions',...
    struct('AcquisitionFunctionName','expected-improvement-plus','kfold',5));
```

Lines 35 to 40 in MLmaterials_L8\Ex_ClassificationSVM.m

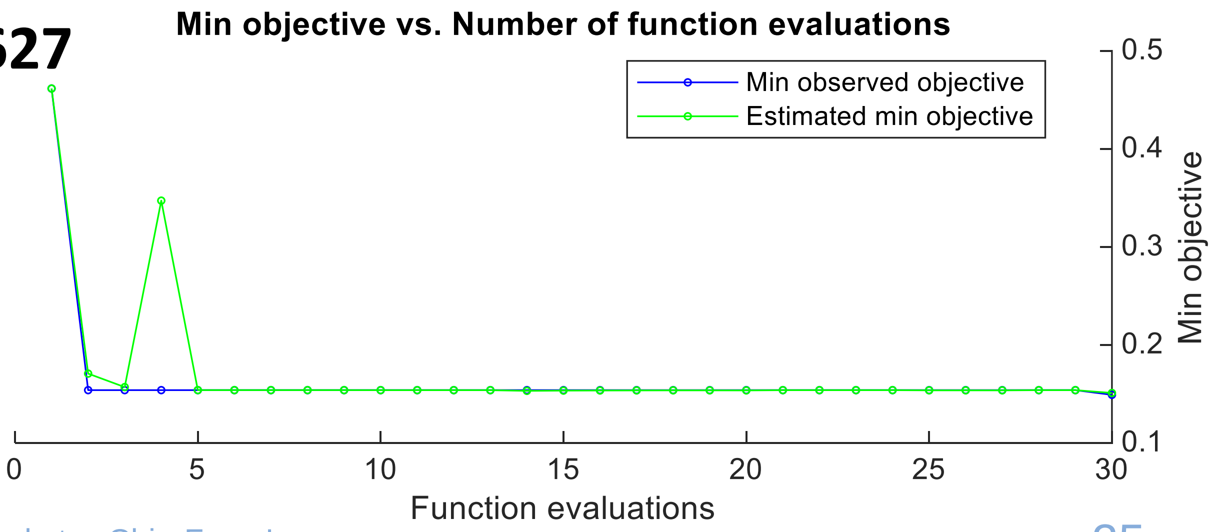
Optimization

Best estimated feasible point (according to models):

BoxConstraint	KernelScale	KernelFunction	PolynomialOrder	Standardize
2.0794	NaN	linear	NaN	false

Estimated objective function value = 0.15094

Estimated function evaluation time = 8.1627



Model Parameters

- save('trainedSVMmodel.mat','svm_model')

svm_model

1x1 ClassificationSVM

Property	Value
BoxConstraints	*208x1 double*
CacheInfo	*1x1 struct*
ConvergenceInfo	*1x1 struct*
Gradient	*208x1 double*
IsSupportVector	*208x1 logical*
Nu	*[ ]*
NumIterations	1000000
OutlierFraction	0
ShrinkagePeriod	0
Solver	'SMO'
Y	*208x1 cell*
X	*208x13 table*
RowsUsed	*[ ]*

Exercise – Classification Tree

```
AHD_predict=predict(svm_model,dataTest);
[cm,order] = confusionmat(dataTest.AHD,AHD_predict)
accurarcy = trace(cm)/sum(cm(:))
```

```
cm =

    45     3
    11    30
```

```
accurarcy =

    0.8427
```

Confusion Matrix		True status	
		Yes	No
Predicted status	Yes	**True Positive** (TP)	False Positive (FP) Type I error
	No	False Negative (FN) Type II error	**True Negative** (TN)

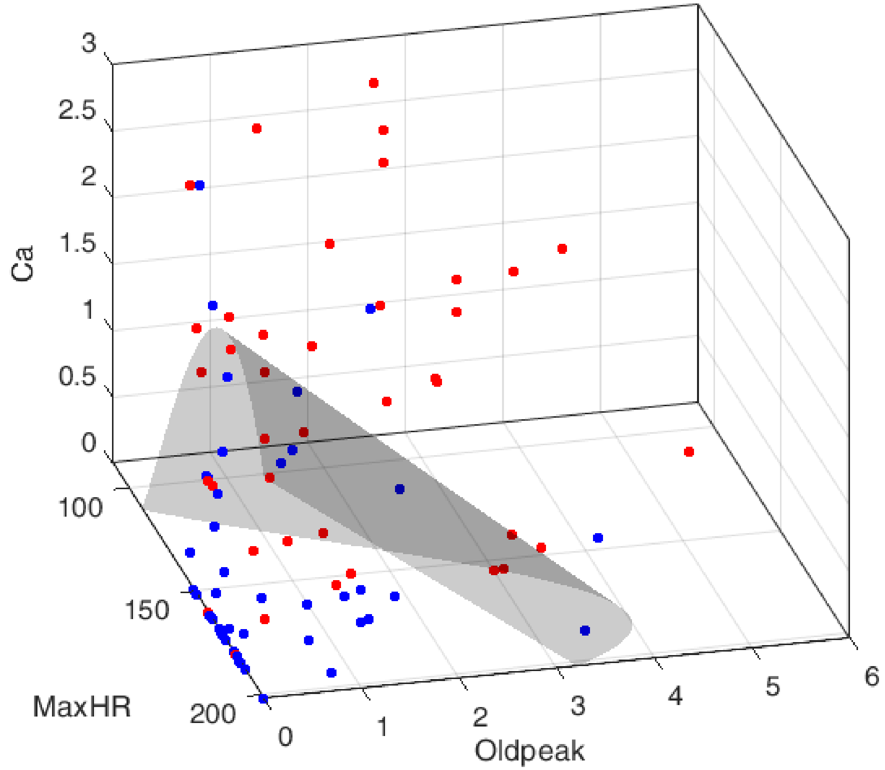
MLmaterials_L8\Ex_ClassificationSVM.m

Display Hyperplane

- We take a 3-D feature dimension as an example (not the best model).

- predictors={'MaxHR','Oldpeak','Ca'};

- [~ , f] = predict(svm_model,xGrid);
- f = reshape(f(:,2), size(x));
- [faces,verts,~] = isosurface(x, y, z, f, 0, x);



MLmaterials_L8\Ex_cSVMhyperplane.m

How can SVM break?

- **Data with lots of error**
  - Hyperplane depends on the few nearest data points (support vectors).
- **Choosing the wrong kernel**
  - Kernel selection is trial and error.
- **Large data sets**
  - Calculating the kernel may become expensive.
- **Feature selection approach may be required.**

- Each of these requires a human in the loop to make judgement calls.

https://youtu.be/-Z4aojJ-pdg

THE END

Contact:

盧家鋒 alvin4016@nycu.edu.tw